REMARKS

The Applicant and the undersigned thank Examiner E. Kiss for his careful review of this application. Consideration of the present application is respectfully requested in view of the above amendments to the specification and claims, and in view of the following remarks.

The Examiner has initially rejected Claims 1-17. Upon entry of this amendment, Claims 1-17 remain pending in this application and new Claims 18-25 have been added. Applicant has amended Claims 1-3, 5-6, 8-11, 13 and 15-17. Claims 1, 11, 18, 20 and 22 are the independent claims of this amended application. In addition, Applicant has amended the specification to present proper usage of trademarks identified by the application.

A marked-up version of the changes made to the specification and claims is not being submitted with this Response because the amendments to the specification and claims listed above have been submitted according to the new procedures entitled, "USPTO ANNOUNCES PROTOTYPE OF IMAGE PROCESSING," and cited in 1265 Off. Gaz. Pat. Office 87 (Dec. 17, 2002) ("Prototype Announcement"). If Examiner Kiss believes the present amendment to be non-responsive because a marked-up version of the claims is not present, the Examiner is invited to contact the undersigned to discuss the matter prior to the Examiner issuing such a non-responsive notice.

Claim Rejections Under 35 U.S.C. § 112

The Examiner has rejected Claims 3, 5, 9 and 16 under 35 U.S.C. § 112, second paragraph, based on the assertion that these claims fail to particularly point out and distinctly claim the subject matter of the claimed invention. Applicants have amended Claims 3, 5, 9 and 16 to address the concerns raised by the Examiner in paragraph 4 of the Official Action.

Claim Rejections Under 35 U.S.C. § 102

The Examiner rejected Claims 1, 5-7, 13-19 under 35 U.S.C. § 102(b) as being anticipated by an article entitled "A Generic Virus Detection Agent on the Internet," *Proceedings of the Thirtieth Hawaii International Conference on System Sciences,*

January 1997, Vol. 4, pages 210-219, by Jieh-Sheng Lee et al. (hereinafter *Lee et al.*) The Applicant respectfully offer remarks to traverse these pending rejections in view of distinctions between the *Lee et al.* reference and the invention as defined by amended independent Claims 1 and 11.

Independent Claims 1 and 11

The rejection of Claims 1 and 11 is respectfully traversed. It is respectfully submitted that the *Lee et al.* reference fails to describe, teach, or suggest the recitations enumerated in amended independent Claims 1 and 11. Specifically, the *Lee et al.* reference fails to describe, teach, or suggest recitations of amended Claim 1, namely (1) initializing a virtual machine comprising a *virtual PC* implemented by software simulating functionality of a central processing unit and memory and a *virtual operating system* simulating functionality of an operating system of a computer system or (2) analyzing behavior of the target program upon completion of virtual execution to identify an occurrence of malicious code behavior based upon an evaluation of a behavior pattern representing information above all functions simulated by the target program during virtual execution. The *Lee et al.* reference also fails to describe, teach, or suggest recitations of amended Claim 11, namely (1) initializing a virtual machine comprising software simulating functionality of a central processing unit and memory and a virtual operating system simulating functionality of a multi-threaded operating system of the computer and (2) terminating the virtual machine upon completion of the virtual execution of the target program, leaving behind a record of the behavior pattern that is representative of operations of the target program with the computer system.

Lee et al. fails to Disclose or Suggest a Virtual Machine Comprising a Virtual PC and a Virtual Operating System as Recited by Independent Claim 1

In paragraph 6 of the Official Action, the Examiner has alleged that the *Lee et al.* reference discloses a "virtual machine comprising software simulating functionality of a central processing unit and memory," as identified in the first paragraph of Section III.C of that reference. While the Applicant acknowledges that the *Lee et al.* reference teaches a software emulator for creating a "virtual system environment" that "acts like a CPU,"

Applicant respectfully traverses the Examiner's assertion that the *Lee et al.* reference teaches a virtual machine as recited by amended independent Claim 1. Significantly, the *Lee et al.* reference fails to disclose, teach, or suggest a virtual machine comprising a virtual PC and a virtual operating system simulating functionality of a *multi-threaded* operating system of a computer, as recited by amended Claim 1. While *Lee et al.* teaches a virtual system environment comprising "virtual registers, virtual memory space, [and] virtual interrupt service routines," this cited reference fails to disclose simulating functionality of a multi-threaded operating system by the use of a *virtual operating system* as defined by amended Claim 1. Consequently, the software emulator of the *Lee et al.* reference, which acts like a CPU, is distinguishable from the structure of the virtual machine recited by amended Claim 1.

In contrast to the software emulator of the *Lee et al.* reference, the invention of amended Claim 1 requires virtually executing a target program within a virtual PC so that the target program interacts only with an instance of the virtual operating system (rather than the operating system of the computer). While the virtual system environment of *Lee et al.* is based on a software emulator that "acts like a CPU," the invention of amended Claim 1 comprises a substantially more complex structure defined by a virtual machine comprising a virtual PC and a virtual operating system. The emulation of service interrupt routines within the virtual system environment of *Lee et al.* is not equivalent to the complexity of simulating the functionality of a multi-threaded operating system, as provided by the virtual operating system of amended Claim 1. Because the *Lee et al.* reference fails to contemplate a multi-threaded operating system environment, the emulated execution of a target file in the virtual system environment of *Lee et al.* can not support interaction of a target program with an instance of a virtual operating system as defined by Claim 1. In view of the foregoing, Applicant respectfully requests that the Examiner withdraw the rejection of independent Claim 1.

Lee et al. fails to Disclose or Suggest Analyzing Behavior of a Target Program upon Completion of Virtual Execution to Identify an Occurrence of Malicious Code Behavior as Recited by Independent Claim 1

The Examiner has asserted in paragraph 6 of the Official Action that Section III.D of the *Lee et al.* reference discloses analyzing behavior of a target program as defined by Claim 1. In contrast, Applicant respectfully submits that the cited section of the *Lee et al.* reference discloses a rule-based virus analyzer that "uses an *event filter* to judge if a received event arising from emulation of a target program is critical." If the event is critical, the event filter in *Lee et al.* sends the event to a *rule matcher* operative to match received events to known virus behavior rules. The virus analyzer in *Lee et al.* will "send a *stop signal* back to the emulator" to terminate execution of the target file if the rule matcher achieves a successful match of a rule with an incoming event.

Significantly, the virus analyzer of the *Lee et al.* reference does not complete execution of the target program before initiating a rule-based analysis of events arising from the execution of that target file. Instead, the analysis of events by the rule matcher in *Lee et al.* is apparently completed in parallel with the emulation of the target program, where only certain events are deemed as significant for matching to the rules of the rules matcher. Moreover, the execution of a target file by the *Lee et al.* virus can be terminated during the course of execution operations if the rule matcher successfully matches a rule with an incoming event generated by file execution. In contrast, the invention of Claim 1, as amended, requires analyzing a pattern of behavior -- not certain discrete events -- for the target program *upon completion* of virtual execution of that target program.

Contrary to the event-driving operation of the virus analyzer in *Lee et al.*, the invention of amended Claim 1 analyzes behavior of a target program based upon a behavior pattern representing information about all functions simulated by the target program during virtual execution. This analysis operation can be performed by the invention of Claim 1, as amended, at completion of virtual execution of the target program because the analysis is based upon an evaluation of a behavior pattern -- rather than a discrete event -- representative of all functions simulated by of the target program during virtual execution. While *Lee et al.* teaches a rules-based analysis that runs in parallel with emulation of the target program, the invention of Claim 1 recites analyzing

behavior of the target program upon completion of virtual execution of that target program. In other words, *Lee et al.* teaches a virus analyzer that stops emulation of the target file upon detection of an event associated with a virus, whereas the invention of Claim 1, as amended, conducts analysis of behavior of the target program at completion of virtual execution of the target program. In view of the clear differences between the invention of amended Claim 1 and the *Lee et al.* reference, Applicant respectfully requests that the Examiner withdraw the rejection of Claim 1 and all claims dependent there from.

Lee et al. fails to Disclose or Teach a Virtual Machine including a Virtual Operating System as Recited by Independent Claim 11

The Examiner has alleged that the *Lee et al.* reference discloses a virtual machine as defined by Claim 11, based upon a citation to the first paragraph Section III.C of the *Lee et al.* reference. As discussed above in connection with independent Claim 1, the *Lee et al.* reference discloses a software emulator that "acts like a CPU" for the execution of a target file within a virtual system environment. Applicant respectfully submits that the virtual system environment of the *Lee et al.* reference is clearly distinguishable from the invention of amended independent Claim 11 because the *Lee et al.* reference fails to disclose, teach, or suggest a virtual machine comprising software simulating functionality of a central processing unit and memory and a virtual operating system.

There exists no suggestion in the *Lee et al.* reference that the virtual system environment includes a virtual operating system capable of simulating functionality of a multi-threaded operating system of a computer, as defined by amended Claim 11. Contrary to the Examiner's allegation, the emulation of interrupt service routines by the software emulator of *Lee et al.* is not equivalent to the virtual operating system defined by Claim 11. The functionality of a multi-threaded operating system, as simulated by the virtual operating system of Claim 11, is implemented by the simulation of operating system data areas and operating system application program interfaces (APIs). The recitation of simulating functionality of a multi-threaded operating system in amended Claim 11 means that a target program executing within the virtual machine interacts with an instance of the virtual operating system rather than an interrupt driven software

emulator. In summary, a software emulator acting like a CPU, as disclosed by the *Lee et al.* reference, is neither identical nor equivalent to the complex software simulations of the virtual machine recited by amended Claim 11. In view of the foregoing, Applicant respectfully submits that the invention of amended Claim 11 is readily distinguishable from the teachings of the *Lee et al.* reference.

The Lee et al. Reference fails to Disclose or Teach Generating a Behavior Pattern for a Target Program to Collect Information about all Functions Simulated By The Target Program as Recited by Independent Claim 11

The Examiner has asserted that the *Lee et al.* reference discloses the use by a virus analyzer of a behavior pattern to indicate the occurrence of malicious code behavior. Applicant respectfully traverses the Examiner's position in connection with independent Claim 11 because the *Lee et al.* reference fails to disclose or teach the generation of a behavior pattern as recited by amended Claim 11. As disclosed in Section III.D of the *Lee et al.* reference, the virus analyzer generates discrete events during execution of a target file and matches each event of significance with built-in rules to detect a virus. Significantly, the virus analyzer of the *Lee et al.* reference terminates all operations by sending a stop signal to the software emulator if an event generated by execution of the target file matches a built-in behavior rule.

In contrast, the invention of amended Claim 11 requires a behavior pattern for the target program to collect information about *all* functions simulated by the target program during the virtual execution. Because the behavior pattern of Amended 11 is generated during virtual execution of the target program within the virtual machine, the behavior pattern represents a collection of information about all functions simulated by the target program. In contrast, the *Lee et al.* system teaches the screening of target program events to determine whether they are critical and warrant analysis by the rules matcher component. Although the rule matcher of the virus analyzer in *Lee et al.* collects certain critical events, derived from execution of the target file, the rule matcher terminates collection of such received events upon the successful match of a rule with an incoming event. In other words, the virus analyzer of *Lee et al.* can not collect information about all functions virtually completed by the target program, as recited for the behavior pattern

of amended Claim 11, because the virus analyzer is designed to filter for critical events and to terminate upon detection of a virus event rather than upon completion of virtual execution of the target program. The behavior pattern of amended Claim 11, in contrast to the event record of *Lee et al.*, provides a record that is representative of operations of the target program with the computer system.

Although *Lee et al.* discloses the collection certain discrete events arising from execution of a target program, the virus analyzer of *Lee et al.* can not collect information about all functions simulated by a target program in the manner recited by independent Claim 11, as amended. The execution information of *Lee et al.* represents a collection of certain critical discrete events generated by emulation of the target program while the behavior pattern of amended Claim 1 comprises information about all functions simulated by the target program during virtual execution. Significantly, the event filter of the virus analyzer in *Lee et al.* will discard certain events if the filter makes a determination that such events are irrelevant, such as screen output routines. By discarding selected events as irrelevant, the *Lee et al.* reference cannot collect information about all functions simulated by the target program, as required for the behavior pattern of amended Claim 11. In view of the forgoing, Applicant respectfully requests that the Examiner withdraw the rejection of independent Claim 11 and all claims there from.

In light of these differences, it is that the *Lee et al.* does not anticipate or render obvious the recitations as set forth in amended independent Claims 1 and 11. Specifically, this reference fails to teach each and every recitation of amended Claims 1 and 11. Accordingly, reconsideration and withdrawal of the initial rejection of Claims 1 and 11, as well as all claims dependent there from, is respectfully requested.

Dependent Claims 2-10 and 12-17

Applicant respectfully submits that the above-identified dependent claims are allowable because the independent claims from which they depend are patentable over the cited references. In view of the foregoing, the Applicant respectfully requests that the Examiner withdraw the pending rejections of dependent Claims 2-10 and 12-17.

Contrary to the Examiner's assertion, the Applicant also respectfully submits that the recitations of these dependent claims are of patentable significance. For example, Applicant submits remarks below in favor of the patentability of selected dependent claims, Claims 3, 6, 8, 13, and 15, to elaborate on the distinctions between the cited prior art and these claims. Moreover, Applicant also reserves the right to submit supplemental remarks, if needed, to further emphasize the distinctions between the inventions of the dependent claims and the cited prior art of record.

Dependent Claim 3

Although the Examiner has alleged that the invention of Claim 3 is disclosed by the "Softworks VBVM" paper, Applicant respectfully submits that the invention of amended Claim 3 is distinguishable from the combination of the *Lee et al.* reference and the Softworks VBVM paper. Dependent Claim 3, as amended, defines a virtual operating system operative to simulate a call of the operating system for the computer system by returning a correct value to the call without completing actual performance of the call. While *Lee et al.* discloses emulated interrupt service routines that return appropriate register values, this reference fails to disclose, teach or suggest the simulation of an application program interface (API) for a multi-threaded operating system, as required for the virtual operating system defined by amended Claim 3. Moreover, the Softworks VBVM paper fails to address the API recitation of amended Claim 3.

Dependent Claims 6 and 8 and Claims 13 and 15

The Examiner has admitted that the *Lee et al.* reference fails to disclose the recitations of dependent Claims 6 and 8 and 13 and 15. Nevertheless, the Examiner has asserted that U.S. Patent No. 5,822,517 to *Dotan* discloses the new behavior pattern recited by dependent Claims 6 and 13 and a method for detecting introduction of malicious code as defined by dependant Claims 8 and 15. Applicant respectfully traverses the rejection of Claims 6 and 13 and amended Claims 8 and 15.

The Examiner has asserted that *Dotan* teaches a "new behavior pattern" as "final state data" that is generated after each execution of a first program. (See *Dotan*, column 7, lines 3-19). A review of the cited section of *Dotan* reveals that the final state data

typically comprises "pertinent information regarding the program, such as program length, header, etc." The pertinent information used to mark the state of the program as a final state in *Dotan* matches the pertinent information of the initial data state if the program has not been infected by a virus.

In contrast to the initial and final data states of *Dotan*, the behavior pattern of dependent Claim 6 is representative of operations of malicious code rather than static characteristics of a program, such as program length or header. Similarly, the behavior pattern of dependent Claim 13 is distinguishable from the final data state of *Dotan* because this behavior pattern is representative of the operation of the target program with the computer system and includes information about all functions completed by the target program during virtual execution. In view of the foregoing, the content of the behavior pattern is readily distinguishable from the type of data contained in the final data state disclosed by the *Dotan* patent. Applicant respectfully requests that the Examiner withdraw the rejection of dependent Claims 6 and 13.

Although the Examiner has asserted that *Dotan* teaches detection of the introduction of malicious code, as recited by dependent Claims 8 and 15, Applicant respectfully submits that the detection of malicious code by the inventions of amended Claims 8 and 15 is distinguishable from the *Dotan* patent. In particular, the *Dotan* patent merely teaches the comparison of the final data state to and the initial data state, as disclosed in column 7, lines 20-26, to determine whether a program has been infected by a virus. This comparison of data states does not include the identification of altered bits indicating an addition of an infection procedure to the modified first program, as defined by amended Claims 8 and 15. As neither the *Lee et al.* reference nor the *Dotan* patent discloses, teaches, or suggests identifying altered bits indicating an addition of an infection procedure to a modified program, as defined by amended Claims 8 and 15, Applicant respectfully requests that the Examiner withdraw this rejection.

New Claims 18-25

Applicant respectfully submits that new Claims 18-25 are patentable for the same reasons outlined above with respect to independent Claims 1 and 11, as amended. Claims 18, 20 and 22 present alternative views of the inventive aspects disclosed by the present

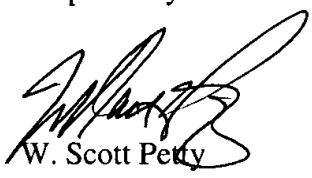
application and are patentable over the prior art of record. Applicant requests that the Examiner pass these claims to allowance in view of the clear distinctions between recitations of Claims 18-25 and the prior art of record.

CONCLUSION

The foregoing is submitted as a full and complete response to the Office Action mailed on March 26, 2003. The Applicant and the undersigned thank Examiner Kiss for his consideration of these remarks. The Applicant has amended the claims and has submitted remarks to traverse the initial rejection of Claims 1-17. The Applicant respectfully submits that the present application is in condition for allowance. Such action is hereby courteously solicited.

If the Examiner believes that there are any issues that can be resolved by a telephone conference, or that there are any formalities that can be corrected by an Examiner's amendment, please contact the undersigned in the Atlanta Metropolitan area (404) 572-2888.

Respectfully submitted,



W. Scott Petty
Reg. No. 35,645

King & Spalding LLP
191 Peachtree Street, NE
Atlanta, Georgia 30303
404.572.4600

K&S File No. 05456-105041
3225708 v1